

6. 実装の違い

[STM32F4DISCOVERY/STM32CubeIDE で、freeRTOS を使う]

2020年8月29日

同じことを、freeRTOS と、μITRON で行った場合の実装の違いを比較してみた。

4つの LED を点滅させるだけのプログラムである。

基本的にはメインのタスクで、100ms ごとに順にトグルを繰り返しているが・・

LD6 はメインのタスクから、ふたつ目のタスクへイベント(シグナル)を飛ばしてトグルさせている。

LD4 は割り込みハンドラからイベント(シグナル)を飛ばして、LD6 と同じふたつ目のタスクで受ける。

freeRTOS(CMSIS API)	μITRON
<pre>void StartTask02(void const * argument) { /* USER CODE BEGIN StartTask02 */ /* Infinite loop */ for(;;) { //HAL_GPIO_TogglePin(LD4_GPIO_Port,LD4_Pin); osDelay(100); HAL_GPIO_TogglePin(LD3_GPIO_Port,LD3_Pin); osDelay(100); HAL_GPIO_TogglePin(LD5_GPIO_Port,LD5_Pin); osDelay(100); //HAL_GPIO_TogglePin(LD6_GPIO_Port,LD6_Pin); osSignalSet(defaultTaskHandle, 1); osDelay(100); } /* USER CODE END StartTask02 */ }</pre>	<pre>void TSK_Test(void) { for(; ;) { //HAL_GPIO_TogglePin(LD4_GPIO_Port,LD4_Pin); wai_tsk(MS(100)); HAL_GPIO_TogglePin(LD3_GPIO_Port,LD3_Pin); wai_tsk(MS(100)); HAL_GPIO_TogglePin(LD5_GPIO_Port,LD5_Pin); wai_tsk(MS(100)); //HAL_GPIO_TogglePin(LD6_GPIO_Port,LD6_Pin); set_flg(FLG_TEST, 1); wai_tsk(MS(100)); } }</pre>
<pre>void StartDefaultTask(void const * argument) { /* init code for USB_HOST */ MX_USB_HOST_Init(); /* USER CODE BEGIN 5 */ HAL_TIM_Base_Start_IT(&htim1); /* Infinite loop */ for(;;) { osEvent event = osSignalWait(0xff, osWaitForever); if(event.value.signals&1) { HAL_GPIO_TogglePin(LD6_GPIO_Port,LD6_Pin); } if(event.value.signals&(1<<1)) { HAL_GPIO_TogglePin(LD4_GPIO_Port,LD4_Pin); } } /* USER CODE END 5 */ }</pre>	<pre>void TSK_Test_2(void) { static UINT event; for(; ;) { wai_flg(&event, FLG_TEST, ~0, TWF_ORW TWF_CLR); if(event&1) { HAL_GPIO_TogglePin(LD6_GPIO_Port,LD6_Pin); } if(event&(1<<1)) { HAL_GPIO_TogglePin(LD4_GPIO_Port,LD4_Pin); } } }</pre>
<pre>void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) { /* USER CODE BEGIN Callback 0 */ /* USER CODE END Callback 0 */ if (htim->Instance == TIM5) { HAL_IncTick(); } /* USER CODE BEGIN Callback 1 */ if (htim == &htim1) { osSignalSet(defaultTaskHandle, (1<<1)); } }</pre>	<pre>void TIM6_DAC_IRQHandler(void) __attribute__((naked)); void TIM6_DAC_IRQHandler(void) { __asm__ volatile ("ent_int();\n" " __HAL_TIM_CLEAR_IT(&htim6, TIM_IT_UPDATE);\n" " set_flg(FLG_TEST, (1<<1));\n" " ret_int();\n" :); }</pre>

<pre>} /* USER CODE END Callback 1 */ }</pre>	
---	--